



Comparison of Neural Network Error Measures for Simulation of Slender Marine Structures

Christiansen, Niels H.; Voie, Per Erlend Torbergson; Winther, Ole; Høgsberg, Jan Becker

Published in:
Journal of Applied Mathematics

Link to article, DOI:
[10.1155/2014/759834](https://doi.org/10.1155/2014/759834)

Publication date:
2014

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Christiansen, N. H., Voie, P. E. T., Winther, O., & Høgsberg, J. B. (2014). Comparison of Neural Network Error Measures for Simulation of Slender Marine Structures. *Journal of Applied Mathematics*, 2014, [759834]. <https://doi.org/10.1155/2014/759834>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Research Article

Comparison of Neural Network Error Measures for Simulation of Slender Marine Structures

Niels H. Christiansen,^{1,2} Per Erlend Torbergson Voie,³ Ole Winther,⁴ and Jan Høgsberg²

¹ DNV Denmark A/S, Tuborg Parkvej 8, 2900 Hellerup, Denmark

² Department of Mechanical Engineering, Technical University of Denmark, 2800 Kongens Lyngby, Denmark

³ Det Norske Veritas, 7496 Trondheim, Norway

⁴ Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kongens Lyngby, Denmark

Correspondence should be addressed to Niels H. Christiansen; niels.christiansen@dnv.com

Received 19 November 2013; Accepted 21 January 2014; Published 2 March 2014

Academic Editor: Ricardo Perera

Copyright © 2014 Niels H. Christiansen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Training of an artificial neural network (ANN) adjusts the internal weights of the network in order to minimize a predefined error measure. This error measure is given by an error function. Several different error functions are suggested in the literature. However, the far most common measure for regression is the mean square error. This paper looks into the possibility of improving the performance of neural networks by selecting or defining error functions that are tailor-made for a specific objective. A neural network trained to simulate tension forces in an anchor chain on a floating offshore platform is designed and tested. The purpose of setting up the network is to reduce calculation time in a fatigue life analysis. Therefore, the networks trained on different error functions are compared with respect to accuracy of rain flow counts of stress cycles over a number of time series simulations. It is shown that adjusting the error function to perform significantly better on a specific problem is possible. On the other hand, it is also shown that weighted error functions actually can impair the performance of an ANN.

1. Introduction

Over the years, oil and gas exploration has moved towards more and more harsh environments. In deep and ultra-deep water installations the reliability of flexible risers and anchor lines is of paramount importance. Therefore, the design and analysis of these structures draw an increasing amount of attention.

Flexible risers and mooring line systems exhibit large deflections in service. Analysis of this behavior requires large nonlinear numerical models and long time-domain simulations [1, 2]. Thus, reliable analysis of these types of structures is computationally expensive. Over the last decades an extensive variety of techniques and methods to reduce this computational cost have been suggested. A review of the most common concepts of analysis is given in [3]. One method that has shown promising preliminary results is a hybrid method which combines the finite element method (FEM) and artificial neural network (ANN) [4]. The ANN

is a pattern recognition tool that based on sufficient training can perform nonlinear mapping between a given input and a corresponding output. The reader may consult, for example, Warner and Misra [5] for a fast thorough introduction to neural networks and their features. The idea with the hybrid method is to first perform a response analysis for a structure using a FEM model and then subsequently to use these results to train an ANN to recognize and predict the response for future loads. As demonstrated by Ordaz-Hernandez et al. [6] an ANN can be trained to predict the deformation of a nonlinear cantilevered beam. A similar approach was used by Hosseini and Abbas [7] when they predicted the deflection of clamped beams struck by a mass. In connection with analysis of marine structures Guarize et al. [8] have shown that a well-trained ANN with very high accuracy can reduce dynamic simulation time for analysis of flexible risers by a factor of about 20. However, a problem with this method is that an ANN can only make accurate predictions

based on sufficiently known input patterns. This means that a network trained on one type of load pattern will have difficulties in predicting the response when the structure is exposed to different types of loading conditions. Recently, a novel strategy for arranging the training data has been proposed by Christiansen et al. [9], where the idea is to select small samples of simulated data for different sea states and collect these in one training sequence. With a proper selection of data an ANN can be trained to predict tension forces in a mooring line on a floating offshore platform for a large range of sea states two orders of magnitude faster than the corresponding direct time integration scheme. It has been shown how computation time, when conducting the simulations associated with a full fatigue analysis on a mooring line system on a floating offshore platform, can be reduced from about 10 hours to less than 2 minutes.

Training of an ANN corresponds to minimizing a pre-defined error function. Several studies on the efficiency of using different objective functions for ANN training have been conducted over the last decades. Accelerated learning in neural networks has been obtained by Solla et al. [10] using relative entropy as the error measure and Hampshire and Waibel [11] presented an objective function called classification figure of merit (CFM) to improve phoneme recognition in neural networks. A comparative study performed by Altincay and Demirekler [12] showed that the CFM objective function also improved the performance of neural networks trained as classifiers for speaker identification. However, these references all consider networks used for classification whereas the one used in this paper is trained to perform regression between time-continuous variables. Hence, the problem studied in this paper calls for different cost functions from those used for neural classifiers.

This study evaluates and compares four different error functions with respect to ANN performance on the fatigue life analysis of the same floating offshore platform as used in [9]. A numerical model of a mooring line system on a floating platform subject to current, wind, and wave forces is established. The model is used to generate several 3-hour time domain simulations at seven different sea states with 2 m, 4 m, . . . , and 14 m significant wave height, respectively. The generated data is then divided into a training set and a validation set. The training set consists of series of simulations at only the sea states with 2 m, 8 m, and 14 m wave height. The remaining part is then left for validation of the trained ANN. The full numerical time integration analysis is carried out by the two tailor-made programs SIMO [13] and RIFLEX [14], while the neural network simulations are conducted by a small MATLAB toolbox.

2. Artificial Neural Network

The artificial neural network is a pattern recognition tool that replicates the ability of the human brain to recognize and predict various kinds of patterns. In the following an ANN will be trained to recognize and predict the relationship between the motion of a floating platform and the resulting tension forces in a specific anchor chain.

2.1. Setting Up the ANN. The architecture of a typical one layer artificial neural network is shown in Figure 1. The ANN consists of an input layer, where each input neuron represents a measured time discrete state of the system. In the present case in Figure 1 the neurons of the input layer represent the six motion components (x, y, \dots) of the floating platform and the previous time discrete anchor chain tension force (T_{past}). The input layer is connected to a single hidden layer, which is then connected to the output layer representing the tension force (T). Two neurons in neighboring layers are connected and each of these connections has a weight. The training of an ANN corresponds to an optimization of these weights with respect to a particular data training set. The accuracy and efficiency of the network depend on the network architecture, the optimization of the individual weights, and the choice of error function used in the applied optimization procedure.

The design and architecture of the ANN and the subsequent training procedure follow the approach outlined in [15]. Assume that the vectors \mathbf{x} , \mathbf{y} , and \mathbf{z} contain the neuron variables of the input layer, output layer, and hidden layer, respectively. The output layer and hidden layer values can be calculated by the expressions

$$\mathbf{y} = \mathbf{W}_O^T \mathbf{z}, \quad \mathbf{z} = \tanh(\mathbf{W}_I^T \mathbf{x}), \quad x_0 \equiv z_0 \equiv 1, \quad (1)$$

where \mathbf{W}_I and \mathbf{W}_O are arrays that contain the neuron connection weights between the input and the hidden layer and the hidden and the output layer, respectively. By setting x_0 and z_0 permanently to one, biases in the data can be absorbed into the input and hidden layer. The tangent hyperbolic function is used as an activation function between the input and the hidden layer. A nonlinear activation function is needed in order to introduce nonlinearities into the neural network. The tangent hyperbolic is often used in networks of this type, which represent a monotonic mapping between continuous variables, because it provides fast convergence in the network training procedure; see [16].

The optimal weight components of the arrays \mathbf{W}_I and \mathbf{W}_O are found by an iterative procedure, where the weights are modified to give a minimum with respect to a certain error function. The updating of the weight components is performed by a classic gradient decent technique, which adjusts the weights in the opposite direction of the gradient of the error function [17]. For the ANN this gradient decent updating can be written as

$$\mathbf{W}_{\text{new}} = \mathbf{W}_{\text{old}} + \Delta \mathbf{W}, \quad \Delta \mathbf{W} = -\eta \frac{\partial E(\mathbf{W})}{\partial \mathbf{W}}, \quad (2)$$

where E is a predefined error function and η is the learning step size parameter. This parameter can either be constant or updated during the training of the ANN. For the applications in the present paper the learning step size parameter is dynamic and will be adjusted for each iteration so that it is increased if the training error is decreased compared to previous iteration steps and reduced if the error increases.

2.2. Error Functions. As mentioned above in Section 2.1 the training of an ANN corresponds to minimizing the associated measure of error represented by the predefined error function

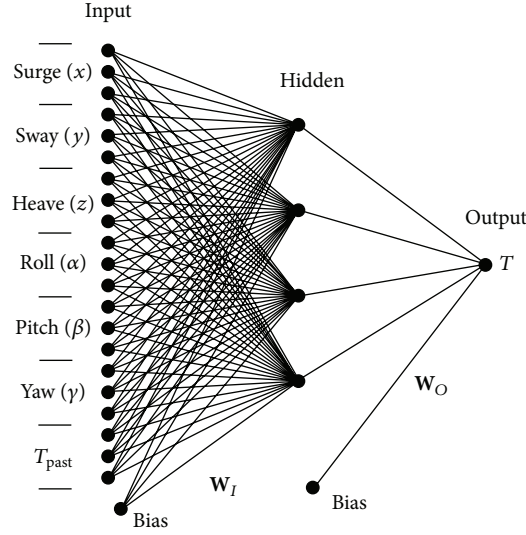


FIGURE 1: Sketch of artificial neural network for predicting top tension force in mooring line.

E. The literature suggests many choices of error functions [16].

The simplest and most commonly used error function in neural networks used for regression is the mean square error (MSE). However, the purpose of the present ANN is to significantly reduce the calculation time for a fatigue analysis of the marine type structure. And since the large amplitude stresses contribute far the most to the accumulated damage of the mooring lines it is of interest to investigate how a different choice of error measure will affect the accuracy and efficiency of the fatigue life calculations. Four different error functions are therefore tested and compared to the full numerical solution obtained by time simulations using the RIFLEX code.

The comparison is based on the so-called *Minkowski-R* error:

$$E = \frac{1}{R} \sum_n \sum_{k=1}^c |y_k(\mathbf{x}_n; \mathbf{W}) - t_{kn}|^R, \quad (3)$$

where y is the scalar ANN output and t is the target value. The classic MSE is seen to be a special case of the Minkowski error with $R = 2$. In many situations the performance and accuracy of the ANN are equally important regardless of the magnitude of the actual output. However, when dealing with analysis of structures this is not always the case. For example, the purpose of the ANN in the present paper is to simulate the top tension force time history in a mooring line, which is subsequently used to evaluate the fatigue life of the line. And since by far the most damage in the mooring line is introduced by large amplitude stress cycles, the ANN inaccuracy on large stresses is much more expensive than errors on small and basically unimportant amplitudes. One way to specifically emphasize large amplitudes is to increase the R -value in (3). Another and more direct way to place additional focus on the importance of large stress amplitudes is by multiplying each term in (3) by the absolute value of

the target values. This yields the following weighted error function:

$$E_w = \frac{1}{R} \sum_n \sum_{k=1}^c |y_k(\mathbf{x}_n; \mathbf{W}) - t_{kn}|^R \cdot |t_{kn}|^R. \quad (4)$$

The performance of a trained ANN is usually measured in terms of the so-called validation error, which is calculated in the same way as the training error but on entirely new data set that has not been part of the network training. This means that when comparing networks that have been trained using different error functions the validation error is the common measure to assess performance in terms of accuracy and computational effort. Obviously the various networks considered in the following could be tested and compared against any of the error functions that the networks have been trained by. But that would definitely favor the particular ANN that has been trained by the specific error function that is chosen as the validation measure. And since the ultimate objective of the ANN is to predict the fatigue life of the mooring line it is appropriate to calculate and compare the accumulated damage in the mooring line caused by all seven sea state realizations previously mentioned in Section 1.

2.3. Network Training. In (2) the steepest decent correction of the weight vector for the training of the network requires the first derivative of the error function E with respect to the weight arrays \mathbf{W} . Differentiation of (3) with respect to the components of the two weight matrices \mathbf{W}_I and \mathbf{W}_O yields

$$\begin{aligned} \frac{dE}{dW_{O,kj}} &= \sum_n |y_k(\mathbf{x}_n; \mathbf{W}) - t_{kn}|^{R-1} z_j, \\ \frac{dE}{dW_{I,ji}} &= \sum_n \left((1 - z_j^2) \sum_{k=1}^c w_{kj} |y_k(\mathbf{x}_n; \mathbf{W}) - t_{kn}|^{R-1} \right) x_i. \end{aligned} \quad (5)$$

These gradients are now inserted into (2) and thereby govern the correction of the weights for each iteration step in

the training procedure. Similar differentiation of the weighted error function (4) yields

$$\begin{aligned} \frac{dE_w}{dW_{O,kj}} &= \sum_n |y_k(\mathbf{x}_n; \mathbf{W}) - t_{kn}|^{R-1} z_j \cdot |t_{kn}|, \\ \frac{dE_w}{dW_{O,ji}} &= \sum_n \left((1 - z_j^2) \sum_{k=1}^c w_{kj} |y_k(\mathbf{x}_n; \mathbf{W}) - t_{kn}|^{R-1} \right) x_i \cdot |t_{kn}|. \end{aligned} \quad (6)$$

The above equations are implemented into the training algorithm and tested for the two power values $R = 2$ and $R = 4$. This gives a total of four different error functions, which in the following will be denoted as

- (i) E_2 : unweighted error function with $R = 2$;
- (ii) E_2^w : weighted error function with $R = 2$;
- (iii) E_4 : unweighted error function with $R = 4$;
- (iv) E_4^w : weighted error function with $R = 4$.

It should be noted that the first case E_2 represents the classic MSE function.

3. Application to Structural Model

The structure used as the basis for the comparison of the different error functions is shown in Figure 2. It consists of a floating offshore platform located at 105 m water depth, which is anchored by 18 mooring lines assembled in four main clusters. The external forces acting on the structure are induced by waves, current, and wind.

3.1. Structural Model. In principal the dynamic analysis of the platform-mooring system corresponds to solving the equation of motion:

$$\mathbf{M}(\mathbf{r}) \ddot{\mathbf{r}} + \mathbf{C}(\mathbf{r}) \dot{\mathbf{r}} + \mathbf{K}(\mathbf{r}) \mathbf{r} = \mathbf{f}(t). \quad (7)$$

In this nonlinear equation \mathbf{r} contains the degrees of freedom of the structural model, and \mathbf{f} includes all external forces acting on the structure from, for example, gravity, buoyancy, and hydrodynamic effects, while the nonconstant matrices \mathbf{M} , \mathbf{C} , and \mathbf{K} represent the system inertia, damping, and stiffness, respectively. The system inertia matrix \mathbf{M} takes into account both the structural inertia and the response dependent hydrodynamic added mass. Linear and nonlinear energy dissipation from both internal structural damping and hydrodynamic damping are accounted for by the damping matrix \mathbf{C} . Finally, the stiffness matrix contains contributions from both the elastic stiffness and the response dependent geometric stiffness.

The nonlinear equations of motion in (7) couple the structural response of the floating platform and the response of the mooring lines. However, the system is effectively solved by separating the solution procedure into the following steps.

First the motion of the floating platform is computed by the program SIMO [13], assuming a quasistatic catenary mooring line model with geometric nonlinearities. The platform response from this initial analysis is subsequently used as excitation in terms of prescribed platform motion in a detailed nonlinear finite element analysis for the specific mooring line with highest tension stresses. The location of the mooring line with largest stresses is indicated in Figure 3. For this specific line the hot-spot with respect to fatigue is located close to the platform and is in the following referred to as the top tension force. From the detailed fully nonlinear analysis performed by RIFLEX the time history of the top tension force at this hot-spot is extracted.

Based on the simulated time histories for both the platform motion and the top tension force an ANN is trained to predict the top tension force in the selected mooring line with the platform response variables as network input. This is considered next in Section 3.2. In [9] a multilayer ANN was trained to simulate the top tension force two orders of magnitude faster than a corresponding numerical model. The training data was set up and arranged so that a single ANN with a single hidden layer could simulate all fatigue relevant sea states and thereby provide a significant reduction in the computational effort associated with a fatigue life evaluation. For clarity the ANN used in this example covers only a few sea states, with different significant wave heights and constant peak period. This gives a compact neural network that is conveniently used to illustrate the influence of changing the error function in the training of the network.

3.2. Selection of Training Data. The ultimate purpose of the ANN is to completely bypass the computationally expensive numerical time integration procedure, which in this case is conducted by the RIFLEX model. This means that the input to the neural network must be identical to the input used for the RIFLEX calculations. In this case the input is therefore the platform motion, represented by the six degrees of freedom denoted in Figure 1 and illustrated in Figure 2. In principle the number of neural network output variables can be chosen freely, and in fact all degrees of freedom from the numerical finite element analysis can be included as output variables in the corresponding ANN. However, the strength of the ANN in this context is that it may provide only the output variable that drives the design of the structure, which in this case is the maximum top tension forces in the particular mooring line. This leads to a very fast simulation procedure, which for a well-trained network provides sufficiently accurate results. Thus, the ANN is in the present case designed and trained to predict the top tension of the mooring line, and the platform motion (six motion components; surge, sway, heave, roll, pitch, and yaw) is, together with the top tension of previous time steps, used as input to the ANN; see Figure 1. This means that the input vector \mathbf{x}_n at time increment n can be constructed as

$$\begin{aligned} \mathbf{x}_n = & \begin{bmatrix} x_t & x_{t-h} & \cdots & x_{t-dh} \end{bmatrix}, \begin{bmatrix} y_t & y_{t-h} & \cdots & y_{t-dh} \end{bmatrix}, \\ & \begin{bmatrix} z_t & z_{t-h} & \cdots & z_{t-dh} \end{bmatrix}, \begin{bmatrix} \alpha_t & \alpha_{t-h} & \cdots & \alpha_{t-dh} \end{bmatrix}, \\ & \begin{bmatrix} \beta_t & \beta_{t-h} & \cdots & \beta_{t-dh} \end{bmatrix}, \begin{bmatrix} \gamma_t & \gamma_{t-h} & \cdots & \gamma_{t-dh} \end{bmatrix}, \\ & \begin{bmatrix} T_{t-h} & T_{t-2h} & \cdots & T_{t-dh} \end{bmatrix}^T, \end{aligned} \quad (8)$$

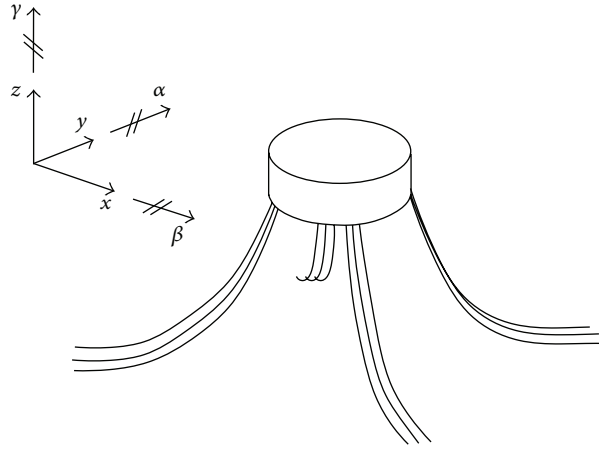


FIGURE 2: Mooring system with floating platform and anchor lines.

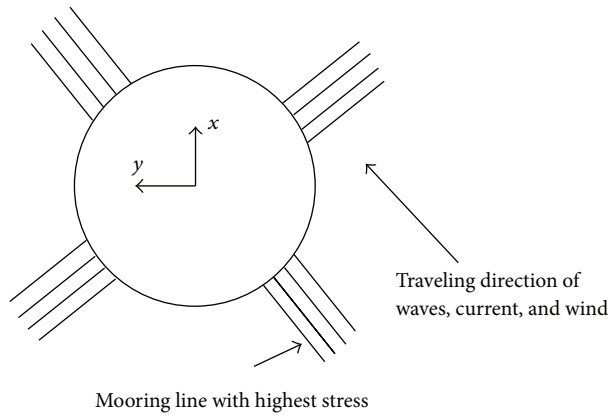


FIGURE 3: Mooring line system (top view).

where $t = nh$ denotes current time, h is the time increment, and d is the number of previous time steps included in the input, that is, the model memory. The corresponding ANN output is the value of the top tension force T_t in the mooring line:

$$y_n = T_t. \quad (9)$$

Since there is only one network output y is a scalar and not a vector as in (1). For the training of the ANN nonlinear simulations in RIFLEX are conducted for sea states with a significant wave height of $H_s = 2$ m, 8 m, and 14 m, respectively. While neural networks are very good at interpolating within the training range, they are typically only able to perform limited extrapolation outside the training range. Thus, the selected training set must contain both the minimum wave height (2 m), the maximum wave height (14 m), and in this case a moderate wave height (8 m) to provide sufficient training data over the full range of interest. With these wave heights included in the training data the ANN is expected to be able to provide accurate time histories for the top tension force for all intermediate wave heights. The seven 3-hour simulation records generated by RIFLEX

are divided into a training set and a validation set. The data that is used for training of the ANN is shown in Figures 4 and 5. Figure 4 shows the time histories for the six motion degrees of freedom of the platform calculated by the initial analysis in SIMO and used as input to both the full numerical analysis in RIFLEX and the ANN training and simulation. Figure 5 shows corresponding time histories for top tension force determined by RIFLEX. The full time histories shown in Figures 4 and 5 are constructed of time series for the three significant wave heights. The first 830 seconds of the training set represent 2 m significant wave height, the next 830 seconds are for 8 m wave height, and the remaining part is then for 14 m wave height.

The SIMO simulations are conducted with a time step size of 0.5 s. In the subsequent RIFLEX simulations the time step must be sufficiently small as to keep the associated Newton-Raphson iteration algorithm stable. In these simulations a time step size of 0.1 s is therefore chosen, which means that the additional input parameters are obtained by linear interpolation between the simulation values from SIMO. For the ANN the time step size h must be chosen so that the network is able to grasp the dynamic behavior of the structure. Therefore, in many cases the ANN is capable of

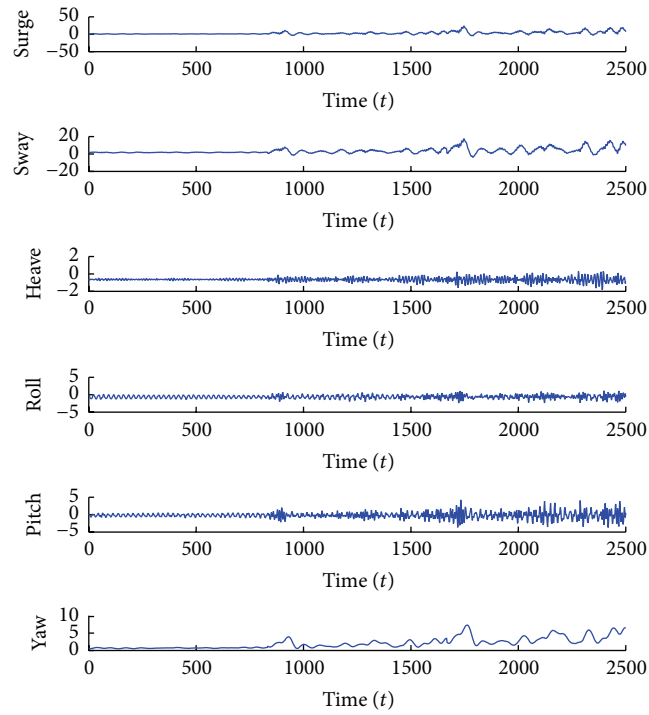


FIGURE 4: Platform motion used as ANN training input data.

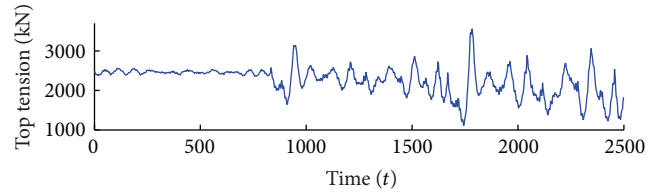


FIGURE 5: Tension force history used as ANN training target data.

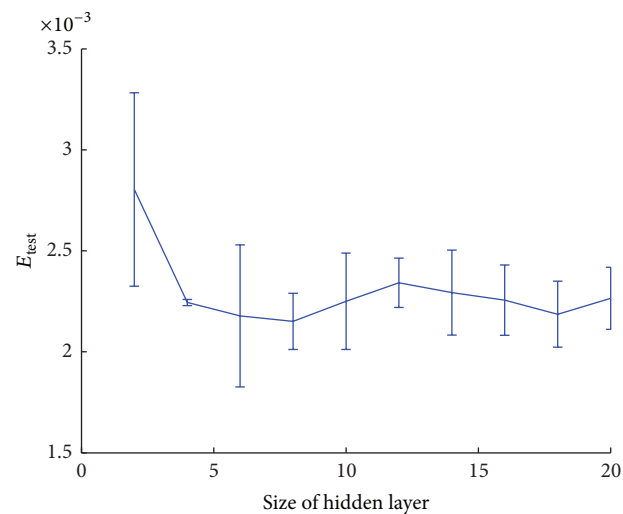


FIGURE 6: Validation error as function of number of neurons in the hidden layer.

handling fairly large time step increments compared to the corresponding numerical models. When using a larger time step in the ANN simulations, for example, by omitting a number of in-between data points, it is possible to reduce the size of the training data set and thereby reduce the computational time used for ANN training and eventually also for the ANN simulation. In the example of this paper a time step size of $h = 2.0$ s is found to yield a good balance between accuracy and computational efficiency, and this time step h is therefore used for the ANN simulations.

3.3. Design of ANN Architecture. In the design of the ANN architecture three variables are investigated: (1) number of neurons in the hidden layer, (2) size of the model memory d , and (3) required amount of training data. When the ANN has been trained and ready for use the network size has no significant influence on the total simulation time and computational effort. The main time consumer is the training part, and time used for training of the network highly depends on network size and the size of the training data set. Hence, it is of great interest to design an ANN architecture that is as compact and effective as possible.

Figure 6 shows a plot of the test error measure E_{test} relative to the number of neurons in the hidden layer of the ANN. In this section, where the three basic ANN variables are chosen the error measure is the mean square error (MSE), corresponding to the E_2 error measure in Section 2.3. The curve in Figure 6 furthermore represents the mean value of the error based on five simulations, while the vertical bars indicate the standard deviation. It is seen from this curve that the performance and the scatter in performance of the trained ANN is lowest when the hidden layer contains four neurons. Therefore, an effective and fairly accurate ANN performance is expected when four neurons in the hidden layer are used in the following simulations.

Figure 7 shows the test error relative to the model memory d , which represents the number of previous time steps used as network input. First of all it is found that including memory in the model significantly reduces the error. However, it is also seen from the figure that an increase of the memory beyond four previous steps implies no significant improvement in the ANN performance. Thus, a four-step memory, that is, $d = 4$, is used in the following numerical analyses.

For the training of any ANN it is always crucial to have a sufficient amount of training data in order to cover the full range of the network and secure applicability with sufficient statistical significance. Figure 8 shows the test error as function of the length of the training data set. As for the parameter studies in Figures 6 and 7 the present curve shows the mean results based on five simulation records. To make sure that a sufficient amount of data is used for the training of the ANN a total simulated record of 2500 s is included, which corresponds to approximately a length of 14 minutes for each of the three sea states. It is seen in Figure 8 that this length of the simulation record is more than sufficient to secure a low error.

The trained ANN is able to generate nonlinear output without equilibrium iterations and hence at an often significantly higher computational pace compared to classic integration procedures with embedded iteration schemes. Figure 9 shows the simulation of the top tension force in the mooring line calculated by the finite element method in RIFLEX and by the trained ANN. The four subfigures in Figure 9 represent the four wave heights that were not part of the ANN training, that is, $H_s = 4$ m, 6 m, 10 m, and 12 m. For these particular simulation records the trained ANN calculates a factor of about 600 times faster than the FEM calculations by RIFLEX.

3.4. Comparison of Error Measures. In the design of the ANN architecture presented above the results are obtained for an ANN trained with the MSE as objective function or error measure. It is in the following conveniently assumed that this ANN architecture is valid regardless of the specific choice of error function. Thus, the various error measures presented in Section 2.3 are in this section compared for the ANN with four neurons in the hidden layer, four memory input variables, and a training length of 2500 s.

As mentioned earlier some of the pregenerated data are saved for performance validation of the trained ANN. These data are used to calculate a validation error E_{test} which is the measure for the accuracy of the trained network. Figure 10 shows the development in the validation error during the network training with all four different error functions present in Section 2.3. It is clearly seen that all four error measures are minimized during training, whereas it is difficult to compare the detailed performance and efficiency of the four different networks based on these curves.

Figure 11(a) summarizes the development of the validation error for the four ANN, but this time the validation error is calculated using the same MSE error measure (E_2) to give a consistent basis for comparison. Thus, the four networks have been trained with four error measures, respectively, while in Figure 11(a) they are compared by the MSE. Even though the networks here are compared on common ground it is still difficult to evaluate how well they will perform individually on a full fatigue analysis. Figure 11(b) illustrates the accuracy of the four networks by showing a close up of a local maximum in the top tension force time history. It is seen that the two unweighted error functions, E_2 and E_4 , perform superiorly compared to the weighted functions. Also the unweighted error measures provide a smaller MSE error in Figure 11(a). This indicates that weighting of the error functions implies no improvement of the performance and accuracy of the ANN.

3.5. Rain Flow Count. The magnitude of the various test error measures is difficult to relate directly to the performance of the ANN compared to the performance of the RIFLEX model. Since these long time-domain simulations are often used in fatigue life calculations an appropriate way to evaluate the accuracy of the ANN is to compare the accumulated rain

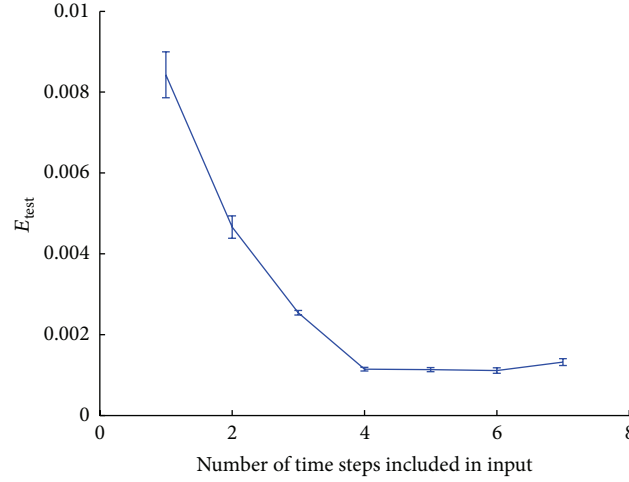


FIGURE 7: Validation error as function of model memory.

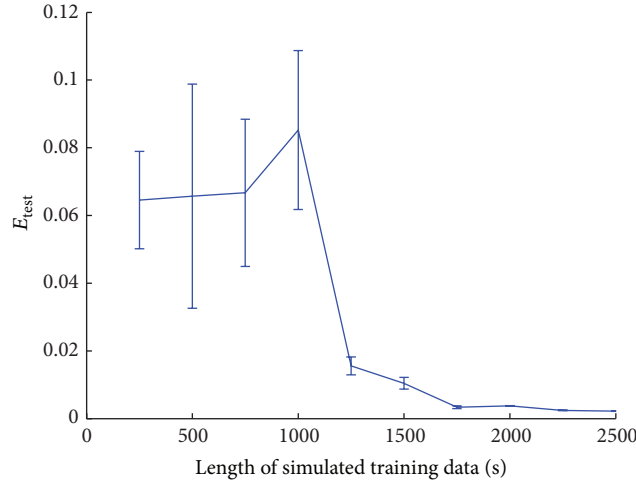


FIGURE 8: Validation error as function of amount of training data.

flow counts of the tension force cycles for each significant wave height. In these fatigue analyses the RIFLEX results are considered as the exact solution. For these calculations the full 3-hour simulations are used and Figure 12 shows the results of the rain flow count of accumulated tension force cycles for each of the significant wave heights. Deviations between RIFLEX and ANN simulations are listed in Table 1. It should be noted that the deviations for the individual seven sea states do not add up to give the total deviation because the individual sea states do not contribute equally to the overall damage. It is seen that the various networks perform very well on all individual sea states and that the best networks thereby obtain a deviation of less than 2% for the accumulated tension force cycles when summing up the contributions from all sea states. This deviation is a robust estimate and is likely to also represent the accuracy of a full subsequent fatigue life evaluation.

It is seen from the rain flow counting results in Figure 12 and Table 1 that the neural networks trained with unweighted

TABLE 1: Deviations on accumulated tension force cycles.

H_s	E_2	E_2^w	E_4	E_4^w
2	-3.6%	-16.3%	7.3%	65.8%
4	-6.9%	-13.0%	4.8%	32.0%
6	-4.4%	-8.0%	2.6%	14.9%
8	-2.7%	-4.8%	-1.5%	8.6%
10	-2.0%	-3.5%	-1.6%	6.9%
12	-1.4%	-3.1%	-1.7%	5.3%
14	-0.8%	-0.3%	2.1%	3.1%
Total	-1.7%	-3.8%	1.6%	6.4%

error function in general perform slightly better than those trained with weighted error functions. Thus, placing a weight on the error function does not seem to have the desired effect in this application concerning the analysis of a mooring line system for a floating platform.

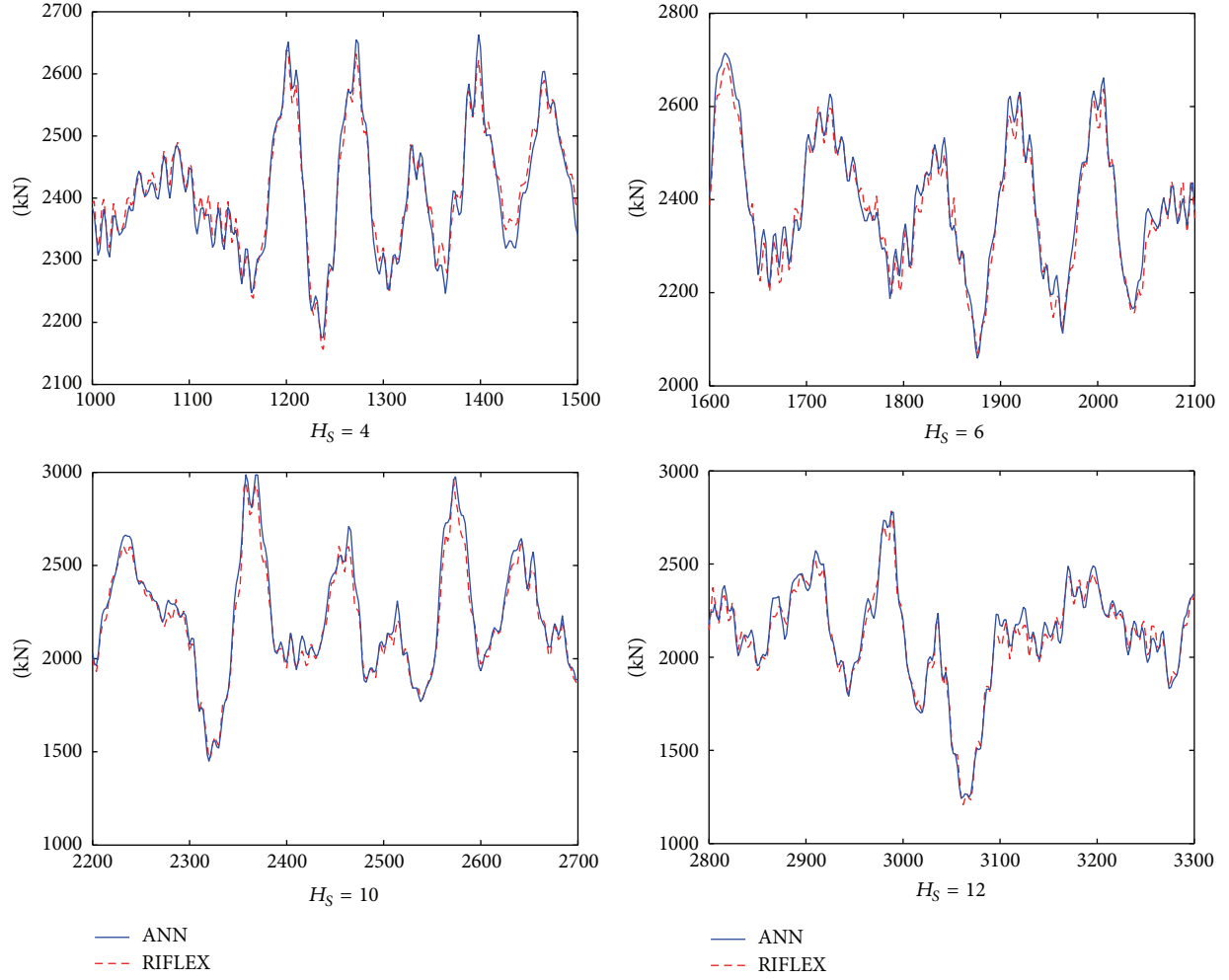


FIGURE 9: Comparison of simulated top tension forces using ANN and RIFLEX for the four wave heights that were not part of the network training.

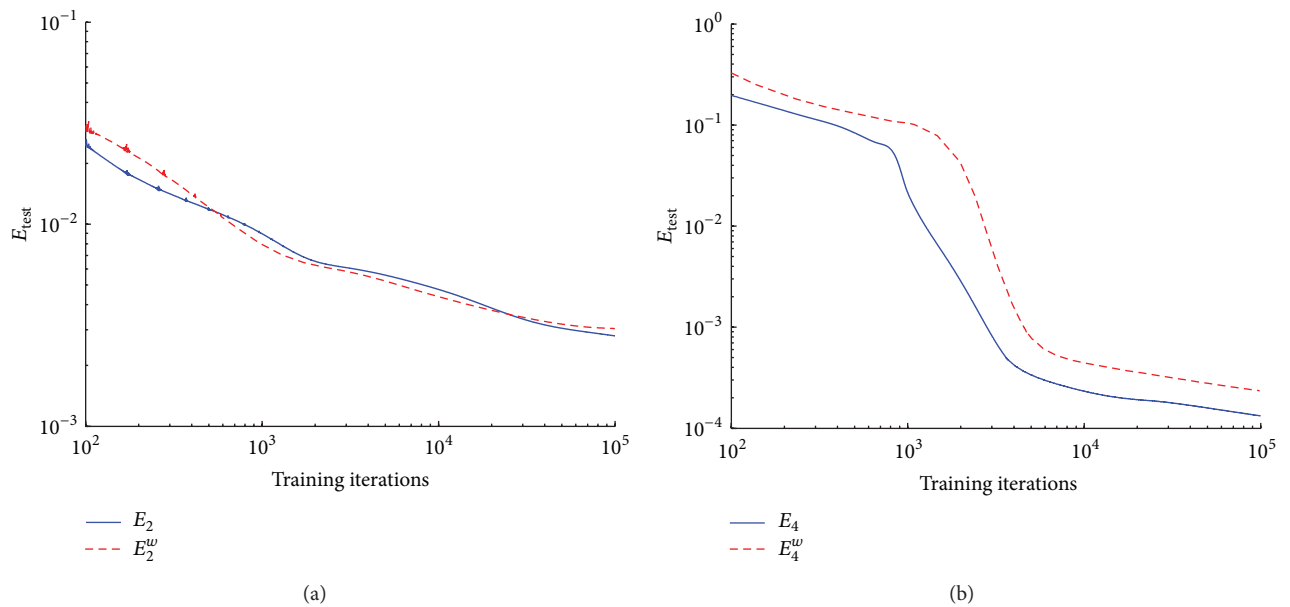


FIGURE 10: Validation error during training. (a) Error functions with $R = 2$; E_2, E_2^w . (b) Error functions with $R = 4$; E_4, E_4^w .

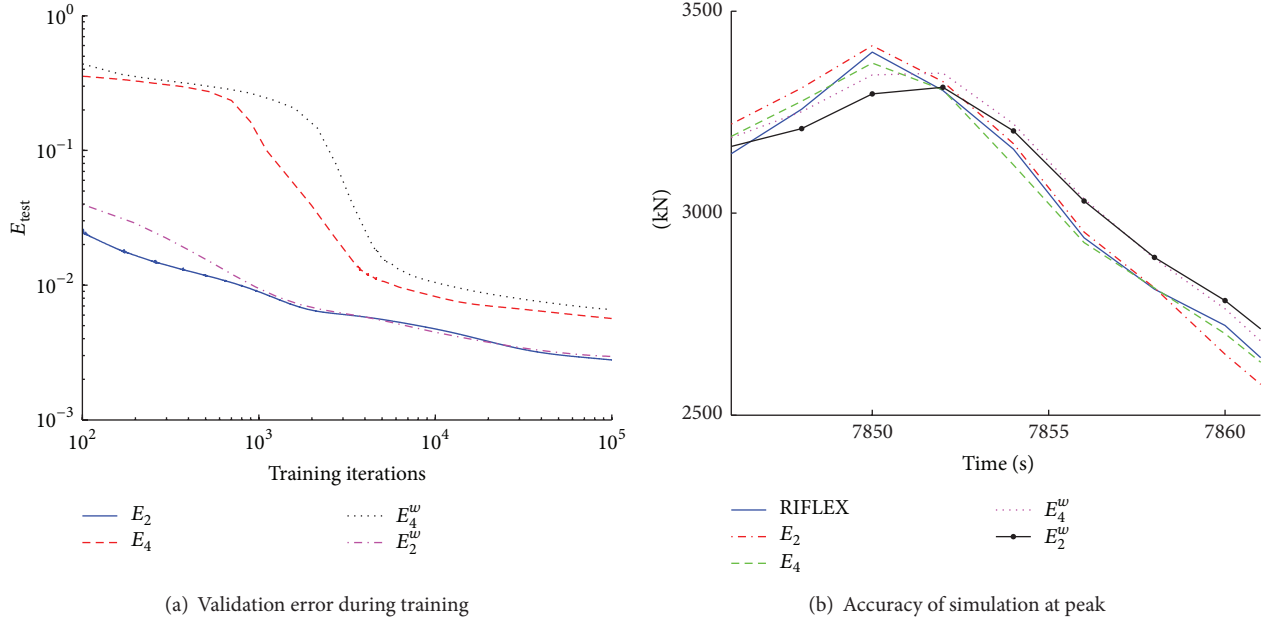


FIGURE 11: Comparison of performance for the four different error functions.

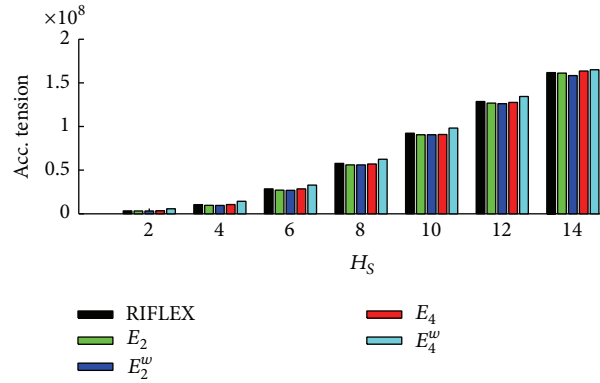


FIGURE 12: Accumulated stress for each sea state—calculated using rain flow counting.

4. Conclusion

It has been shown how a relatively small and compact artificial neural network can be trained to perform high speed dynamic simulation of tension forces in a mooring line on a floating platform. Furthermore, it has been shown that a proper selection of training data enables the ANN to cover a wide range of different sea states, even for sea states that are not included directly in the training data. In the example presented in this paper it is clear that weighting the error function used to train an ANN in order to emphasize peak response does not improve the network performance with respect to accuracy of fatigue calculations. In fact, the ANN appears to perform worse when trained with the weighted error function. On the other hand it appears that increasing the power of the error function from two to four provides a slight improvement to the performance of the trained ANN.

However, the idea of a weighted error function seems to reduce the ANN performance. So apparently focusing on the high amplitudes seems to deteriorate the low amplitude response more than it improves the response with large amplitudes.

As a conclusion the Minkowski error with $R = 4$ is interesting for the mooring line example in more than one aspect. It provides more focus on the large amplitudes and improves the ANN slightly. Furthermore, the second derivative of the E_4 is fairly easy to determine, which makes this objective function suitable for several network optimizing schemes, such as Optimal Brain Damage (OBD) and Optimal Brain Surgeon (OBS), that are based on the second derivative of the error function. Network optimization is, however, not considered further in this paper but will be subject of future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] DNV, "Offshore Standard DNV-OS-F201—Dynamic Risers," Det Norske Veritas, October 2010.
- [2] DNV, "Recommended Practice DNV-RP-F204—Riser Fatigue," Det Norske Veritas, July 2005.
- [3] M. H. Patel and F. B. Seyed, "Review of flexible riser modelling and analysis techniques," *Engineering Structures*, vol. 17, no. 4, pp. 293–304, 1995.
- [4] H. Adeli, "Neural networks in civil engineering: 1989–2000," *Computer-Aided Civil and Infrastructure Engineering*, vol. 16, no. 2, pp. 126–142, 2001.
- [5] B. Warner and M. Misra, "Understanding neural networks as statistical tools," *The American Statistician*, vol. 50, no. 4, pp. 284–293, 1996.
- [6] K. Ordaz-Hernandez, X. Fischer, and F. Bennis, "Model reduction technique for mechanical behaviour modelling: efficiency criteria and validity domain assessment," *Journal of Mechanical Engineering Science*, vol. 222, no. 3, pp. 493–505, 2008.
- [7] M. Hosseini and H. Abbas, "Neural network approach for prediction of deflection of clamped beams struck by a mass," *Thin-Walled Structures*, vol. 60, pp. 222–228, 2012.
- [8] R. Guarize, N. A. F. Matos, L. V. S. Sagrilo, and E. C. P. Lima, "Neural networks in the dynamic response analysis of slender marine structures," *Applied Ocean Research*, vol. 29, no. 4, pp. 191–198, 2007.
- [9] N. H. Christiansen, P. E. T. Voie, J. Høgsberg, and N. Sødahl, "Efficient mooring line fatigue analysis using a hybrid method time domain simulation scheme," in *Proceedings of the 32nd ASME International Conference on Ocean, Offshore and Arctic Engineering (OMAE '13)*, vol. 1, 2013.
- [10] S. A. Solla, E. Levin, and M. Fleisher, "Accelerated learning in layered neural networks," *Complex Systems*, vol. 2, no. 6, pp. 625–639, 1988.
- [11] J. B. Hampshire and A. H. Waibel, "Novel objective function for improved phoneme recognition using time-delay neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 2, pp. 216–228, 1990.
- [12] H. Altincay and M. Demirekler, "Comparison of different objective functions for optimal linear combination of classifiers for speaker identification," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, vol. 1, pp. 401–404, May 2001.
- [13] MARINTEK, *Simo Theory Manual*, Sintef, Trondheim, Norway, 2009.
- [14] MARINTEK, *Riflex Theory Manual*, Sintef, Trondheim, Norway, 2008.
- [15] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, USA, 2006.
- [16] C. M. Bishop, *Neural Networks for Pattern Recognition*, The Clarendon Press, Oxford, UK, 1995.
- [17] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, Chichester, UK, 2nd edition, 1987.

